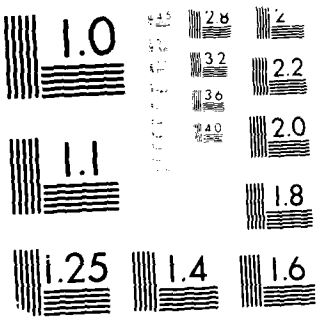
AD-A190 633 SUPERCOMPUTER ENVIRONMENT(U) ILLINOIS UNIV AT URBANA 1/1
30 OCT 87 AFOSR-TR-87-1968 F49620-86-C-0136

UNCLASSIFIED F/G 12/5 NL



MICROCOPY RESOLUTION TEST CHART

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

(2)

REPORT DOCUMENTATION PAGE

DTIC FILE COPY

AD-A190 633

DTIC ELECTE JAN 14 1988 S D

1b. RESTRICTIVE MARKINGS	
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
4. PERFORMING ORGANIZATION REPORT NUMBER(S) 701	5. MONITORING ORGANIZATION REPORT NUMBER(S) AFOSR-TR- 87-1968
6a. NAME OF PERFORMING ORGANIZATION The Board of Trustees of the University of Illinois	6b. OFFICE SYMBOL (If applicable)
6c. ADDRESS (City, State, and ZIP Code) 506 S. Wright Street Urbana, IL 61801	7a. NAME OF MONITORING ORGANIZATION AFOSR/NM
	7b. ADDRESS (City, State, and ZIP Code) AFOSR/NM Bldg 410 Bolling AFB DC 20332-6448
8a. NAME OF FUNDING/SPONSORING ORGANIZATION AFOSR	8b. OFFICE SYMBOL (If applicable) NM
	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER AFOSR F49620-86-C-0136
8c. ADDRESS (City, State, and ZIP Code) AFOSR/NM Bldg 410 Bolling AFB DC 20332-6448	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 61102F; PROJECT NO. 2304; TASK NO. A3; WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

Supercomputer Environments

12. PERSONAL AUTHOR(S)

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Annual Technical	FROM 10-1-86 TO 9-30-87	1987, Oct. 30	

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This summary covers goals and first year accomplishments of the Faust project, a research effort to develop a software engineering environment of supercomputer application development. The goal of Faust is to provide a user-friendly workstation of integrated development tools that caters to a broad range of user expertise. Faust comprises individual efforts in the area of user interface tools, expert systems, parallel program debugging, performance evaluation, symbolic processing, error analysis, and graphics. The Faust project is additionally supporting efforts in the areas of fluid and structural dynamics. Overall project goals are outlined, followed by a summary of first contract year's accomplishments that were presented to the Faust project review team on August 6, 1987.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION
22a. NAME OF RESPONSIBLE INDIVIDUAL Maj. John P. Thomas	22b. TELEPHONE (Include Area Code) (202) 767-5026
	22c. OFFICE SYMBOL NM

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete.

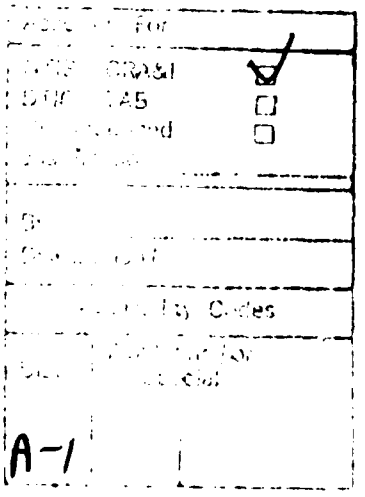
SECURITY CLASSIFICATION OF THIS PAGE

AFOSR-TR- 87-1968

Annual Technical Report

Supercomputer Environments

Contract No. F49620-86-C-0136



SUPERCOMPUTER ENVIRONMENTS

First Year Goals

- Design of a unified user interface for scientific programming environments
- Development of an interactive editor/restructurer for Fortran programs
- Development of new techniques for the debugging of parallel programs
- Research new techniques for parallel ray tracing
- Research new techniques for performance data collection and rendering
- Improve code reusability through the use of electronic consultants which help locate numerical library routines
- Research new techniques for using high resolution graphics for presentation of CFD and structural dynamics applications
- Research the parallelization of Lisp
- Improve current technology with respect to automatic error analysis of scientific programs
- Research new techniques for the automatic optimization of scientific programs

Overview

The goal of this research is the development of a portable integrated supercomputer programming environment for scientific applications. Development efforts are focused on the implementation of an effective set of tools to support the areas of program design and construction, debugging, performance analysis, and output visualization.

Development Tools

During the first year, emphasis has been placed on the development of an overall design as well as a core set of user–interface utilities and conventions. These core utilities are complete and will form the foundation for future development tools.

An expert system for aiding programmers in the location of numerical library routines has been completed and integrated into Faust. This system queries the programmer about his needs and directs him to the appropriate optimized kernels. The system may also ascertain some pieces of information through automatic analysis of data files.

Debugging is currently supported through two tools. One is a conventional symbolic breakpoint debugger that is built on the standard DBX debugger supplied with the Concentrix Operating System from Alliant. This debugger has been enhanced to support a more powerful front–end language and is currently being modified to support extensions for monitoring and altering the execution of parallel programs. The other debugging tool is an automatic analyzer. The Execution Analysis SYstem (EASY) locates nondeterminism in programs by automatically inserting software trace instrumentation. The trace output is analyzed after execution to locate actual or potential race conditions. Future work in this area will concentrate on the development of features to facilitate monitoring and probing of parallel programs as well as the incorporation of a graphic user interface.

A preliminary facility for performance evaluation has been completed. Through post–execution analysis, this tool supports the examination of run–time behavior for several processors through both static (Gantt–style) and dynamic ("replay" of execution) presentations. Future work will focus on the development of new methods of collecting and presenting performance data.

Error Analysis

Integration of automatic error analysis into the Faust environment has been started. Each step of a computation, whether it is an input or arithmetic operation, may be subject to some error counted by the loss of units in the last place of the result. Although the exact error is not known, an upper bound is available. By taking these bounds into account, one is able to evaluate the sensitivity of the outputs with respect to the data (problem condition) and to the arithmetic (algorithm condition) for given inputs. By solving an optimization problem, a measure of the stability of the algorithm is also derived.

In order for this analysis to be possible, the trace of the floating–point and input operations is needed. For this purpose, an interface module has been developed as part of the Cedar Fortran preprocessor (option –z). Therefore, an instrumented program file may be generated given the original program. In this file, floating–point operators have been replaced by Fortran functions which add to the trace when it is called. The pass operates in parallel, assigning a processor to each line. As a result, when the instrumented file is run for the given inputs, the computational graph is generated and used, in turn, as input to the main part of the error analysis programs.

Small–scale programs have been run successfully through the preprocessor and analyzed, giving the theoretically expected results. At the current stage, the memory and computational requirements for the analysis can be enormous even for algorithms of moderate size. Hence, the next task will be to modify the algorithms used in order to utilize parallel and vector computational resources efficiently. Work is also being done to allow for the representation of errors due to the intrinsic mathematical functions.

Graphics

The CSRD graphics effort exists to develop tools required for displaying results from supercomputer simulations and displaying the progress of simulations. The overall direction is to unify these tools and techniques into the Faust environment.

A set of routines to allow cooperating programs on two machines to interact has been completed. Testing of these network routines is complete. They allow, for example, Alliant code to send raw data with very high-level graphics instructions to the Pixar Image Computer (PIC). The network interface will be used for sending commands and responses and for synchronization. The routines are being incorporated into a CFD graphics code to allow a simulation-time (as opposed to real-time) display of a program's progress.

The latest version of the commercial ray tracing package TRACER has been installed and tested. It currently runs in Fortran on the Alliant and its output pictures are displayed using the PIC. TRACER is being merged with ray tracing software written at CSRD. This will ultimately improve TRACER's lighting model, increase the number of available geometric primitives, and allow application of parallel programming techniques already applied to a simple ray tracer at CSRD.

Various techniques which introduce scalar (MIMD) and vector (SIMD) parallelism into the ray tracing environment on a vector multiprocessor have been explored. Initially, a simple ray tracing package was used for experimentation with some parallelization techniques. Then the tracer was modified to incorporate certain algorithm enhancements which were designed to reduce the overall run time of the program because of decreased computation and increased processor utilization. The runs of various implementations were done on the Alliant FX/8 computer and the PIC was used to view the image files.

List of Professional Personnel

1. Dr. David J. Kuck
2. Dr. Duncan H. Lawrie
3. Dr. Ahmed H. Sameh
4. Dr. David Padua
5. Dr. Dennis Gannon
6. Dr. Daniel Reed
7. Dr. Mehdi Harandi
8. Dr. Thomas Huang
9. Dr. Efstratios Gallopoulos
10. Vince Guarna
11. Allen Malony
12. Allan Tuchman
13. David Jablonowski
14. Yogesh Gaur
15. Michael Berry

Air Force Office of Scientific Research
AFOSR F49620-86-C-0136

Publications

[PaGL87] David A. Padua, Vincent A. Guarna Jr. and Duncan H. Lawrie. *Supercomputer Programming Environments.* **Subm. for publ. to Symp. on Parallel Computations and Their Impact on Mechanics, Boston, MA**, December 1987.

Submitted Papers

[Guar87] Vincent A. Guarna Jr. *VPC - A Proposal for a Vector Parallel C Programming Language.* **Subm. for publ. to IEEE Software**, June 1987.

Internal Reports

[Malo87] Allen D. Malony. "Concurrency Efficiency User's Manual", CSRD Report No. 675, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., June 1987.

[Malo87] Allen Malony. "High Resolution Process Timing User's Manual", CSRD Report No. 676, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., June 1987.

[Malo87] Allen D. Malony. "Program Profiling in Cedar", CSRD Report No. 654, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., March 1987.

DATE
FILMED
4 8